



Hak cipta dan penggunaan kembali:

Lisensi ini mengizinkan setiap orang untuk mengubah, memperbaiki, dan membuat ciptaan turunan bukan untuk kepentingan komersial, selama anda mencantumkan nama penulis dan melisensikan ciptaan turunan dengan syarat yang serupa dengan ciptaan asli.

Copyright and reuse:

This license lets you remix, tweak, and build upon work non-commercially, as long as you credit the origin creator and license it on your new creations under the identical terms.

BAB III

METODE DAN PERANCANGAN SISTEM

3.1 Metodologi Penelitian

Metodologi penelitian yang akan digunakan dalam perancangan *roguelike 2D Top-Down Shooting* dengan menggunakan algoritma *Drunkard Walk* sebagai berikut.

a. Studi Literatur

Pada tahap ini akan dilakukan beberapa riset terkait perancangan dan pembangunan permainan ini berupa pendalaman konsep teori dari *Drunkard Walk*, *Game Design*, dan *Roguelike*.

b. Perancangan Permainan

Pada tahap perancangan permainan, akan dilakukan sebuah dokumentasi untuk menjelaskan desain permainan secara rinci, termasuk *flowchart*, asset yang digunakan, dan penjelasan konsep *game design* dari permainan yang dibuat.

c. Pembangunan Permainan

Pada tahap pembangunan permainan, *game* diciptakan dengan menggunakan *game engine* Unity 2018.4.2f1 64-bit dengan target *platform* utama berupa komputer dengan sistem operasi Windows 10.

d. Pengujian Permainan

Pengujian dilakukan dengan *playtesting* dari beberapa responden yang memainkan dan menginstall permainan secara acak dan *developer* sebagai *tester*. Jumlah responden yang dipilih secara acak sebanyak 30 responden. Responden yang dipilih secara acak akan mengisi sebuah kuesioner berdasarkan sistem *GUESS*

untuk mengumpulkan dan mengukur tingkat kepuasan pemain terhadap permainan yang diuji.

e. Evaluasi Permainan

Pada tahap akhir ini, permainan akan dilakukan beberapa analisa terkait pengumpulan data dari kuesioner *GUESS* yang dikumpulkan dengan menggunakan metode skala *likert* untuk mengetahui keberhasilan permainan yang dibuat dengan menggunakan *Drunkard Walk* terhadap *user experience* dari pemain.

3.2 Perancangan Aplikasi

Berikut adalah perancangan untuk aplikasi yang dibangun.

3.2.1 Struktur Permainan

Judul Permainan: NO ESCAPE

Mengambil kondisi permainan yang bersifat *survival* dikarenakan banyak permainan *roguelikes* yang bersifat *permadeath*.

Formal elements yang terdapat di dalam permainan dapat dijabarkan sebagai berikut.

1. Players

Single Player Game, Permainan dimainkan oleh satu pemain, tidak terdapat fitur *multiplayer*.

2. Objectives

mengalahkan semua musuh yang ada di satu *level* suatu permainan agar dapat mengakses menuju *level* selanjutnya.

3. Procedures

- a. Pemain menjalankan aplikasi permainan dan masuk ke menu utama. untuk memulai permainan, pemain harus memilih opsi *start*.
- b. Ketika opsi *start* dipilih, pemain akan masuk ke bagian *cutscene* cerita dari suatu permainan.
- c. Setelah pemain memilih opsi *next* pada *cutscene*, permainan yang dibangun dengan algoritma *Drunkard Walk* dan kombinasi *A* Pathfinding* telah selesai dibangun. Pemain diwajibkan untuk mengalahkan semua musuh yang ada di dalam *level* permainan. Ketika semua musuh dikalahkan, pemain akan melanjutkan permainan pada level selanjutnya.
- d. Terdapat 3 jenis *Boss* pada permainan ini, ketiga *Boss* memiliki jumlah darah yang berbeda dan memiliki kekuatan spesial berupa membangkitkan *minion*.
- e. Apabila nyawa pemain menyentuh angka 0, pemain akan mendapatkan pesan dan *Game Over* ditampilkan. Pemain diberikan dua opsi untuk memulai ulang permainan atau kembali pada *main menu*.
- f. Setelah menyelesaikan level terakhir permainan, pemain akan mendapatkan *cutscene* terakhir lalu pemain akan dikembalikan ke halaman utama.
- g. Ketika pemain memilih opsi *menu*, pemain akan langsung masuk ke *scene main menu* dan pemain akan memulai permainan pada level awal dan kembali pada prosedur kedua.

4. Rules

- a. Pemain menggunakan *keyboard* dan *mouse* untuk memainkan permainan.
- b. Pemain harus mengalahkan semua musuh untuk dapat mengakses *level* selanjutnya.

- c. Pemain mengalahkan musuh dengan menyerang musuh sampai nyawa musuh habis.

5. Resources

- a. *Player Health*: menunjukkan nyawa pemain yang mempengaruhi ukuran senter pemain. Semakin berkurang jumlah darah pemain, maka ukuran senter pemain akan berkurang.
- b. *Ammunition*: menunjukkan jumlah amunisi yang ada pada senjata pemain untuk menembak.
- c. *Food*: menunjukkan jumlah makanan yang dimiliki oleh pemain yang mempengaruhi kecepatan pemain ketika bergerak. Semakin berkurang jumlah makanan, pergerakan pemain akan semakin melambat.

6. Conflict

Menyelesaikan setiap *level* yang di-*generate* dan tetap hidup dengan musuh yang mencoba untuk menyerang pemain hingga nyawa pemain mencapai 0.

7. Boundaries

Pemain berinteraksi dengan control *keyboard* dan *mouse*. *Game Space* yang diciptakan berbentuk *dungeon* dan pemain akan selalu berada di dalam *dungeon*.

8. Outcome

- a. *Statistics*: menunjukkan statistika dari pemain yang telah memainkan permainan seperti jumlah musuh yang dibunuh.

Berikut adalah *Dramatic Elements* yang terdapat di dalam permainan, antara lain.

1. Challenge

Menghancurkan seluruh musuh yang ada di suatu level permainan.

2. Play

Rule-Based Play. Pemain memiliki aturan dan petunjuk ketika memainkan permainan.

3. Character

Survivor yang berbekal sebuah senjata.

4. Premise

Seorang *survivor* yang selamat dari *post-apocalypse* yang merenggut nyawa ketiga adiknya. *Survivor* ingin membalaskan dendam terhadap kehilangan adiknya dengan mengalahkan semua *zombie*.

5. Story

Suatu hari di suatu daerah yang telah didominasi oleh *zombie*, terdapat satu satunya *survivor* yang selamat dari *apocalypse*. Berbekal pistol dan sepucuk gambar berupa ingatan sang *survivor* dengan adiknya, dia ingin membalaskan dendam kematian adik-adiknya. Namun untuk membalaskan dendamnya, dia harus menghadapi musuh-musuh yang ada seperti *zombie*, bandit yang bersindiket sebagai dalang terjadinya *apocalypse*, dan juga *husk* yaitu *zombie* yang telah berevolusi.

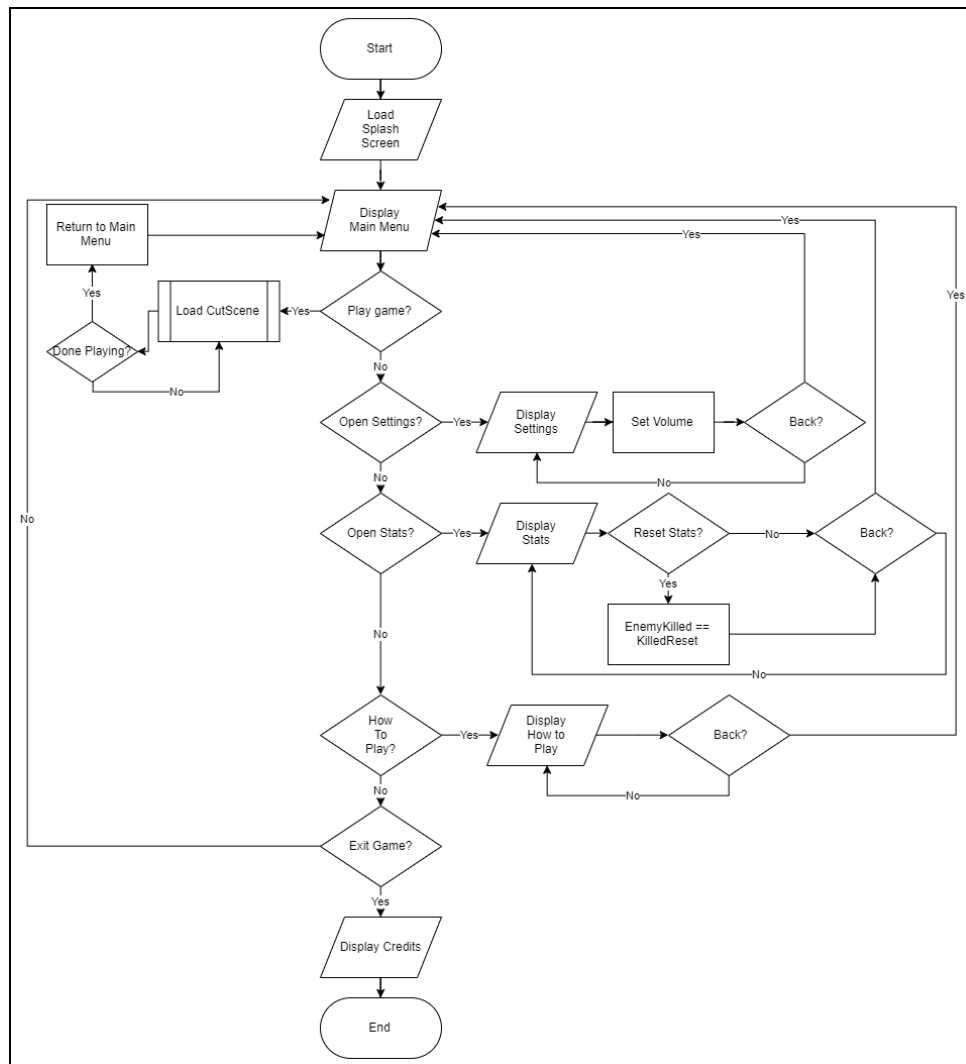
6. World Building

Permainan ini memiliki tiga jenis *setting* yang berbeda untuk memberikan atmosfir yang berbeda kepada pemain.

1. Desert Cave (Level 1 sampai dengan Level 5)
2. Hidden Tribes (Level 6 sampai dengan Level 10)
3. Cursed Laboratory (Level 11 sampai dengan Level 15)

3.2.2 Flowchart Penggunaan Asset

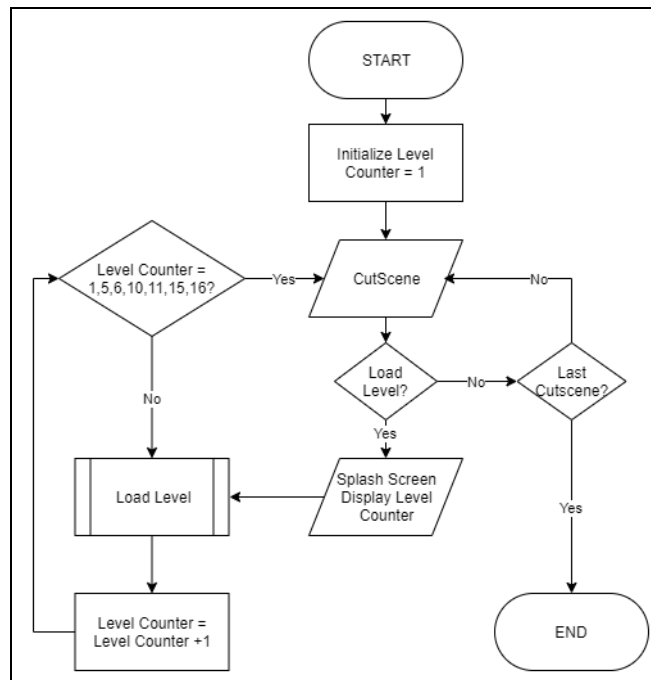
Berikut adalah tampilan desain *Flowchart* untuk permainan. Dimulai pada Gambar 3.1 yang menunjukkan alur menuju *main menu* pada awal permainan ketika pemain memulai permainan.



Gambar 3.1 *Flowchart* Sistem *Main Menu*

Gambar 3.1 menunjukkan alur dari permainan secara umum, ketika pemain memulai permainan akan diperlihatkan sebuah *splash screen* selama 5 detik. Setelah *splash screen* selesai ditampilkan, pemain akan ditampilkan *main menu*.

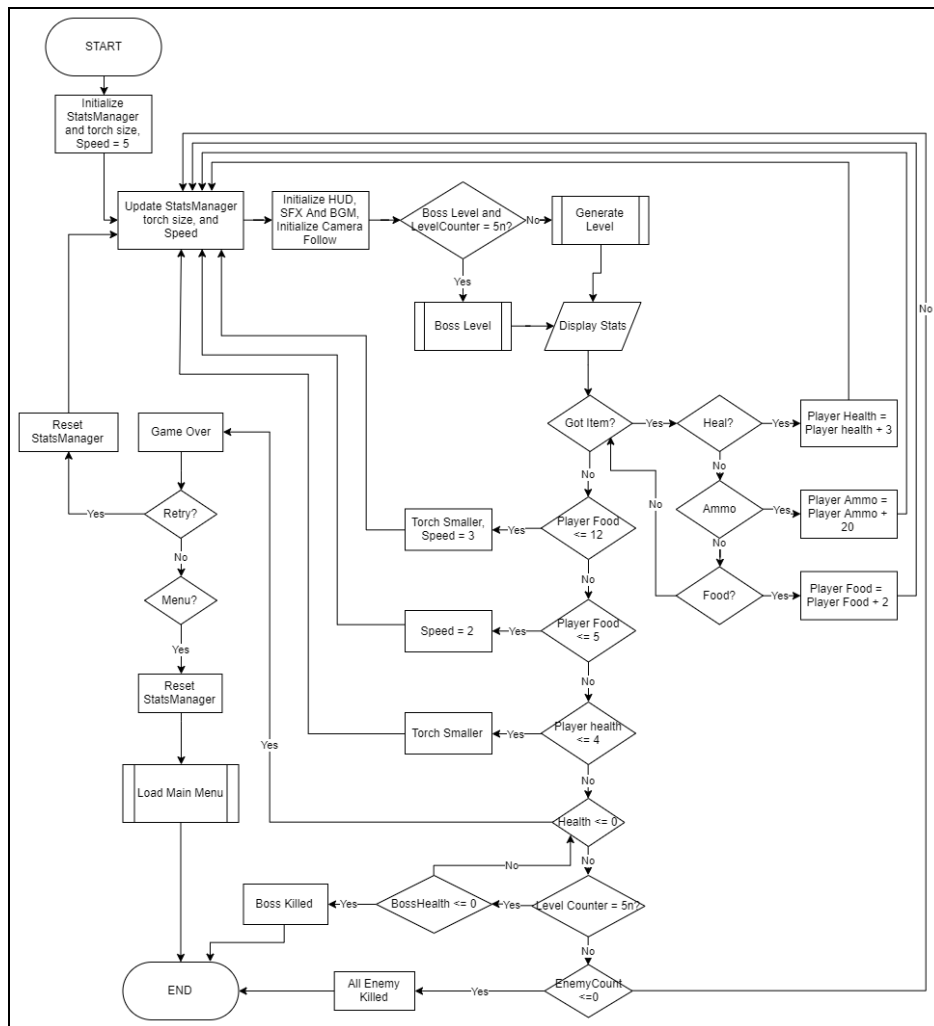
Pada *main menu* ditampilkan 5 jenis opsi seperti opsi untuk memulai permainan yang mengakses *cutscene* terlebih dahulu, opsi *settings* untuk mengatur suara, *statistics* dengan logo list untuk menunjukkan jumlah musuh yang dihancurkan dan dapat diatur ulang, opsi *how to play* untuk menampilkan informasi *HUD display* dan kontrol pemain, dan opsi terakhir untuk keluar dari permainan yang menunjukkan *credits*.



Gambar 3.2 *Flowchart Sistem Cutscene*

Gambar 3.2 menunjukkan mekanisme *cutscene* sebelum memulai permainan. pada bagian ini dimunculkan *cutscene* awal dari permainan berupa *prologue* cerita. Setelah melihat *cutscene* akan ditampilkan sebuah *display* level permainan selama 2 detik dan menyimpan *value* dari *level counter*. Jika *level counter* sudah memasuki 5, 6, 10, 11, dan 15, maka permainan akan memunculkan *cutscene* sebelum *boss level* akan muncul dan setelah boss pada level permainan hancur akan ditampilkan *cutscene* setelah mengalahkan boss dan menambah *level*

counter 1 tingkat. Jika *level counter* tidak memasuki kriteria dari *level counter*, maka permainan akan memunculkan *level* permainan tanpa menampilkan *cutscene* lagi dan jika pada suatu *level* permainan pemain menghancurkan semua musuh, maka *level counter* bertambah 1 tingkat. Jika *cutscene* yang didapat merupakan *cutscene* terakhir, maka permainan selesai dan kembali pada menu utama yang telah digambarkan pada Gambar 3.1.



Gambar 3.3 Flowchart Gameplay Scene

Gambar 3.3 menunjukkan alur *gameplay* setelah membaca *cutscene*. pada *flowchart* ini permainan akan memuat informasi pemain berupa *stats player* dan

juga ukuran *torch* yang kemudian di *update* secara terus menerus. Setelah mendapatkan informasi, permainan memuat HUD berupa informasi *stats* pemain, lalu memuat seluruh audio yang diperlukan pada permainan.

Setelah inisialisasi, permainan mengecek kondisi permainan berdasarkan *level counter* yang digunakan pada Gambar 3.2. Jika *level counter* memasuki kelipatan 5, maka akan memuat *boss level* yang dijelaskan pada Gambar 3.5. Namun apabila *level counter* tidak memasuki kelipatan 5, maka akan memuat sebuah *level* yang dihasilkan secara acak.

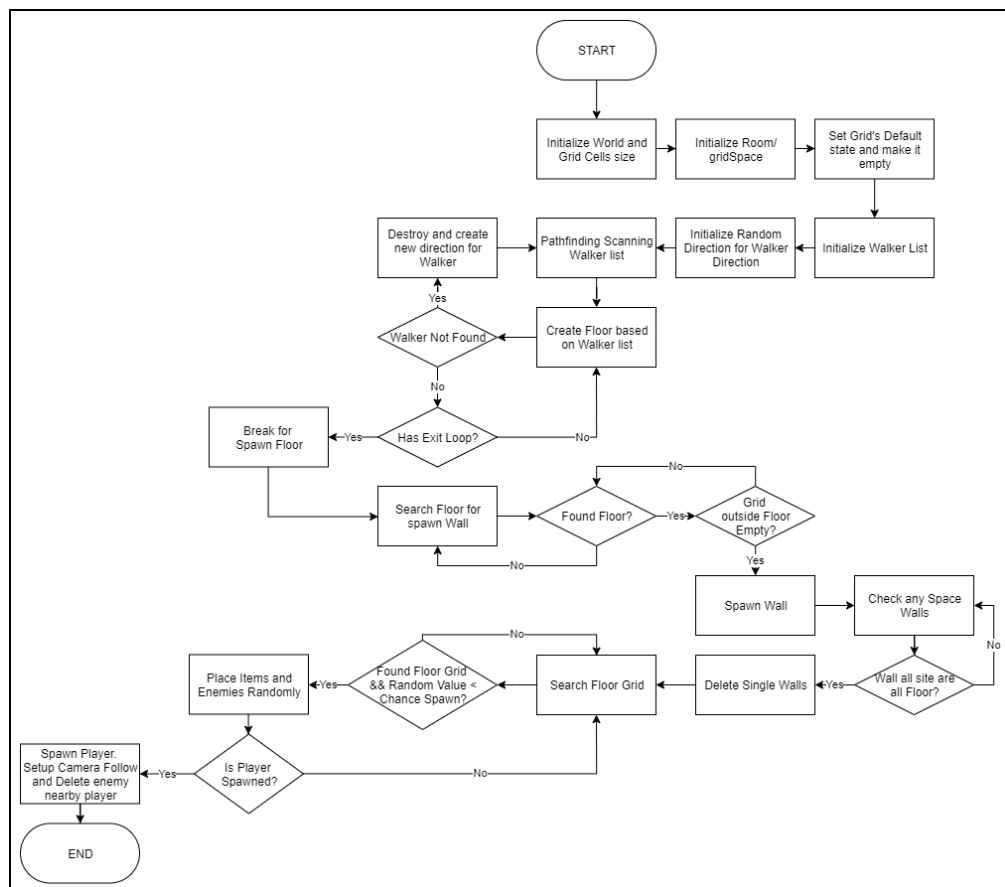
Setelah penentuan *level*, kamera pada permainan akan memuat dalam permainan untuk mencari keberadaan pemain dan menampilkan *stats* pemain setelah memuat HUD. Jika *level counter* memasuki kelipatan 5, maka kamera cukup untuk mencari keberadaan pemain. Namun jika *level counter* tidak memasuki kelipatan 5, maka kamera akan mencari keberadaan pemain berdasarkan hasil dari *generate level*.

Pemain memiliki *stats* berupa *health*, *ammo*, dan *food*. *Health* adalah jumlah darah dari pemain yang mempengaruhi ukuran *torch*. *Ammo* merupakan jumlah amunisi pada pemain ketika ingin menembak. *Food* merupakan jumlah makanan yang mempengaruhi kecepatan pergerakan pada pemain dan juga ukuran *torch* ketika jumlahnya dibawah dari 13.

Pemain juga bisa memanfaatkan *items* yang dihasilkan dari *generate level* apabila *items* itu ada pada *level permainan*. jika pemain mendapatkan *items* berupa *heal* maka darah akan bertambah sebanyak 3. Jika pemain mendapatkan *items* berupa amunisi maka amunisi akan bertambah sebanyak 20 dan jika pemain mendapatkan *items* berupa makanan jumlah *food* pada pemain bertambah sebanyak

2. Apabila jumlah darah pemain habis, panel *game over* ditampilkan. Pada panel *game over* disediakan dua opsi untuk mengulang permainan dari *level* awal dan juga opsi untuk kembali ke *main* menu.

Setelah mengecek jumlah darah pemain, selanjutnya mengecek ulang *level counter*. Jika *level counter* memasuki kelipatan 5, maka permainan akan mengecek kondisi *boss*. Jika darah *boss* habis, maka *level* permainan berakhir dan lanjut pada *level* selanjutnya. Namun jika *level counter* tidak memasuki kelipatan 5, maka permainan akan mengecek jumlah musuh dalam permainan. Apabila jumlah musuh habis, maka pemain dapat melanjutkan *level* selanjutnya. Semua data yang diambil akan disimpan dan di *update* pada *Status Manager*.



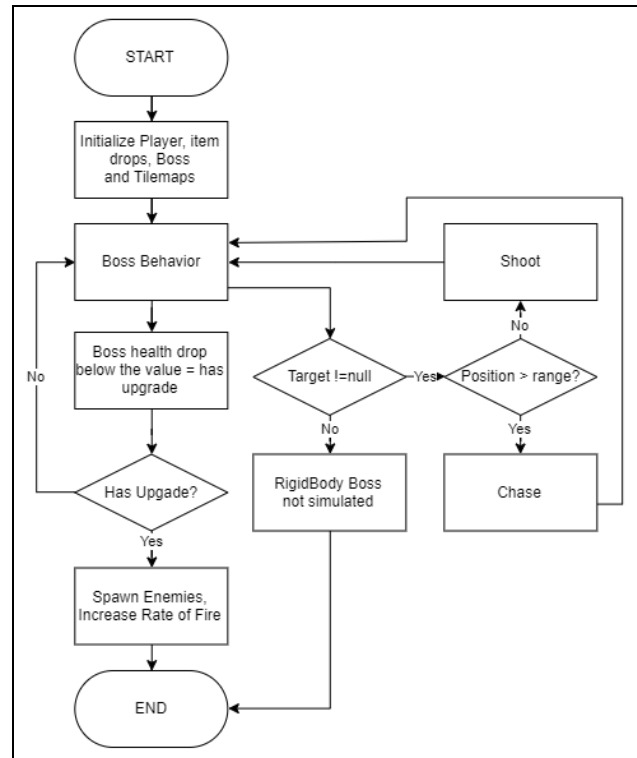
Gambar 3.4 Flowchart Level Generator

Pada Gambar 3.4 menunjukkan cara kerja dari algoritma *Drunkard Walk* untuk menghasilkan sebuah *level* permainan secara acak. Pada implementasi algoritma ini, dimulai pada *setup* algoritma yang mengatur ukuran ruangan permainan dalam skala *grid* dan juga mengatur pergerakan pada *walker* yang menciptakan ruang pada *grid*. *Walker* yang telah selesai di *generate* lalu dimasukan kedalam *walker list* untuk dijadikan ruangan permainan.

Setelah *Walker* selesai berjalan pada *grid*, *setup* yang telah berjalan dan menghasilkan *gridspace* akan diisi objek berupa lantai permainan yang mengecek kembali apabila tidak ditemukan *walker*. Jika *walker* tidak ditemukan, maka *walker* yang dibuat akan dihancurkan dan dibuat *walker* baru dengan arah yang berbeda dari *walker* sebelumnya.

Setelah lantai berhasil dibuat semua dari *grid*, kemudian dinding akan diciptakan dari hasil lantai yang dibuat. Apabila terdapat *single wall*, dinding itu dihapus dan digantikan oleh lantai. Setelah dinding berhasil dibuat, seluruh *items* dan *enemy* akan diciptakan berdasarkan ruang *grid* yang dihasilkan, kemungkinan dari objek untuk dapat dihasilkan secara acak.

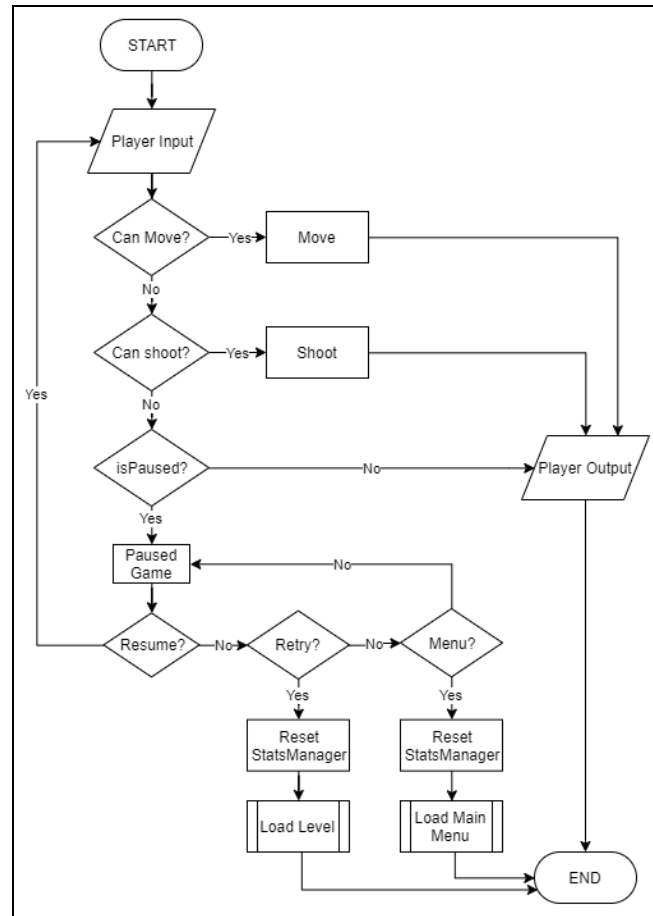
Setelah seluruh *items* dan *enemy* dibuat, algoritma ini akan menghasilkan *player spawn* yang diletakkan secara acak dan posisi kamera yang mengikuti pemain. Apabila tidak ditemukannya objek *player*, maka algoritma ini akan mencari kembali kemungkinan *spawn* yang ada dari *gridspace* lantai yang telah dibuat.



Gambar 3.5 *Flowchart Boss Level*

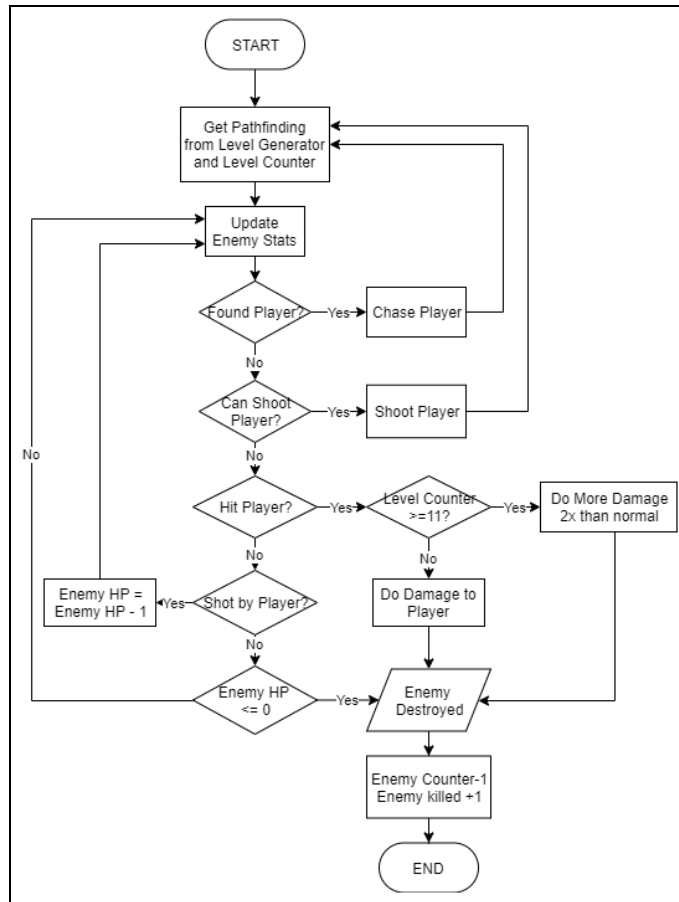
Pada Gambar 3.5 menunjukkan mekanisme dari *boss* pada *boss level* yang tidak menggunakan algoritma *Drunkard Walk*. Pada umumnya *boss level* memiliki alur permainan yang mirip dengan *gameplay*, namun *grid* yang dibuat melalui *tilemaps* dan tidak menggunakan algoritma.

Pada mekanisme *boss*, memiliki dua opsi untuk pemain yaitu opsi untuk mengejar pemain apabila diluar jangkauan *boss* dan opsi untuk menembak apabila berada di jangkauan. Ketika darah *boss* memenuhi kriteria untuk mendapatkan *upgrade*, *boss* akan menghidupkan musuh dari area dan meningkatkan kecepatan menembak kepada pemain. Apabila *boss* tidak menemukan pemain, maka mekanisme *boss* akan berhenti.



Gambar 3.6 *Flowchart Player Control*

Pada Gambar 3.6 menunjukkan mekanisme kendali pada pemain. Pemain diberikan dua mekanisme yaitu bergerak dan menembak. Jika pemain menekan tombol *escape* pada *keyboard*, maka panel *pause* akan muncul dan memberikan tiga opsi yaitu opsi untuk melanjutkan permainan, opsi untuk memulai permainan dari awal, dan opsi untuk kembali pada menu utama. Seluruh efek pergerakan yang berubah dijelaskan pada Gambar 3.3.



Gambar 3.7 Flowchart Enemy Patrol

Pada Gambar 3.7 menunjukkan mekanisme pada *enemy*. *Enemy* memiliki mekanisme *Pathfinding* yang menggunakan unity project dari Aron Granberg (Granberg, 2019). Apabila *enemy* menemukan pemain, maka *enemy* akan mengejar pemain. Terdapat opsi untuk menembak, namun hanya dimiliki oleh *enemy* pada *level stage* 6 hingga 10. Apabila *enemy* menabrak *player*, maka darah pemain akan berkurang dan dapat berkurang dua kali lipat apabila *enemy* yang dihadapi dari *level stage* 11 hingga 15. Setelah *enemy* menabrak *player*, objek yang menabrak akan dihancurkan.


Jika *enemy* terkena tembakan dari pemain, maka darah *enemy* akan berkurang satu dan jika darah *enemy* habis, maka objek *enemy* akan dihancurkan

serta mengurangi jumlah musuh pada permainan dan menambahkan jumlah musuh yang dihancurkan pada *statistik* menu utama. Jika jumlah musuh telah habis, maka akan dilanjutkan pada *level* selanjutnya yang telah dijelaskan pada Gambar 3.2.










3.2.3 Penggunaan Aset

Berikut adalah penggunaan aset untuk perancangan dan pembangunan permainan.

Tabel 3.1 Daftar Aset

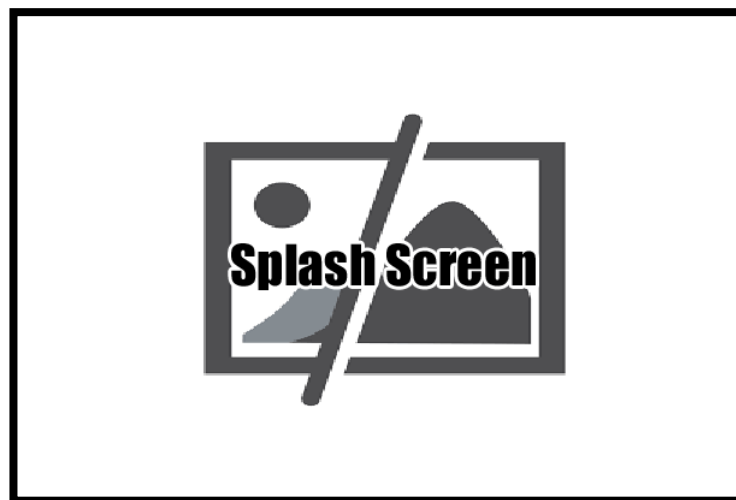
Gambar	Deskripsi	Sumber
	Karakter Pemain dalam sebuah permainan.	Aset Pribadi
	Musuh pada level permainan awal yang hanya bisa mengejar.	Aset Pribadi
	Musuh pada level permainan yang bisa menembak dan mengejar pemain.	Aset Pribadi
	Musuh pada level permainan akhir yang lebih sakit dan lebih cepat namun tidak bisa menembak.	Aset Pribadi
	Objek untuk menambah darah pemain.	Aset Pribadi
	Objek untuk menambah amunisi pemain.	Aset Pribadi

Tabel 3.2 Daftar Aset (lanjutan)

Gambar	Deskripsi	Sumber
	Objek untuk menambah makanan pemain.	Aset Pribadi
	Tanah pada level permainan.	Aset Pribadi
	Senjata yang digunakan pada permainan.	Aset Pribadi
	A-star Pathfinding Object	Granberg, 2019
	<i>Background Music</i> pada <i>Main Menu</i> .	Author: joeBaxterWebb Source: https://opengameart.org/content/what-a-lovely-apocalypse
	<i>Background Music</i> ketika <i>Game Over</i> .	Author: Machine Source: https://opengameart.org/content/crypt
	Kumpulan <i>audio</i> yang digunakan pada <i>Gameplay</i> .	Seluruh efek audio yang berasal dari opengameart.org dan dapat digunakan untuk kepentingan riset dan komersial
	<i>Background Music</i> pada <i>Gameplay</i> .	Author: onderwish Source: https://freesound.org/people/onderwish/sounds/468407/
	<i>Background Music</i> pada <i>Boss Battle</i> .	Author: Kevin Macleod Source: https://incompetech.com/music/royalty-free/index.html?isrc=usuan1100305

3.2.4 Desain Mockup

Berikut adalah tampilan desain *Mockup* untuk permainan. Dimulai pada Gambar 3.8 yang menunjukkan tampilan *splash screen* pada awal permainan ketika pemain memulai permainan.



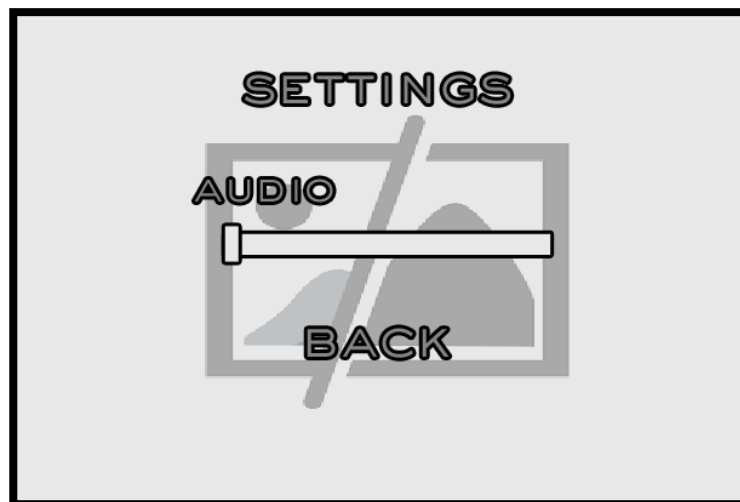
Gambar 3.8 Tampilan *Splash Screen*

Pada Gambar 3.8 menggambarkan desain *mockup* tampilan utama dari permainan. *Splash screen* menunjukkan nama pencipta dari permainan. Gambar ini merupakan gambaran dari desain *flowchart* pada Gambar 3.1.



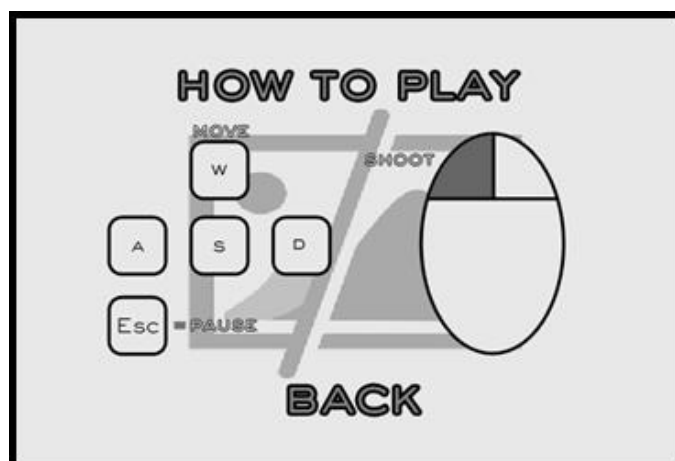
Gambar 3.9 Tampilan awal pada *Main Menu*

Gambar 3.9 merupakan desain *mockup* setelah *mockup splash screen* selesai ditampilkan. Pada tampilan awal *Main Menu* ditampilkan opsi untuk memulai permainan dari awal, mengakses *settings*, melihat cara bermain, melihat statistik dan keluar dari permainan.



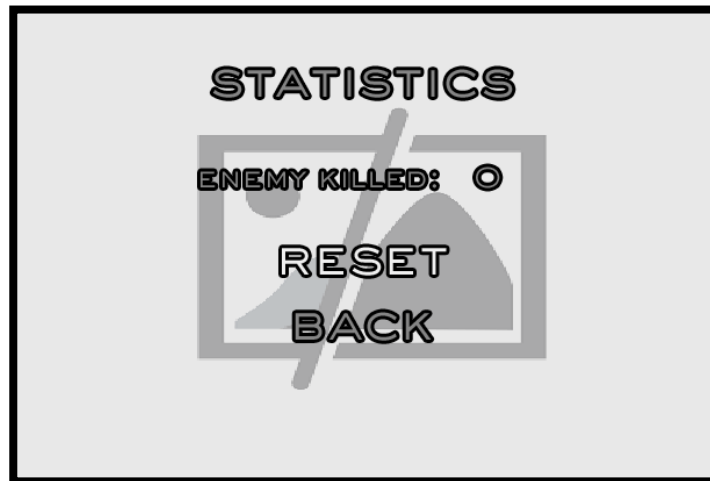
Gambar 3.10 Tampilan pada *Settings*

Gambar 3.10 merupakan tampilan permainan ketika memilih opsi *Settings*. Terdapat opsi *slider* untuk mengatur suara secara global.



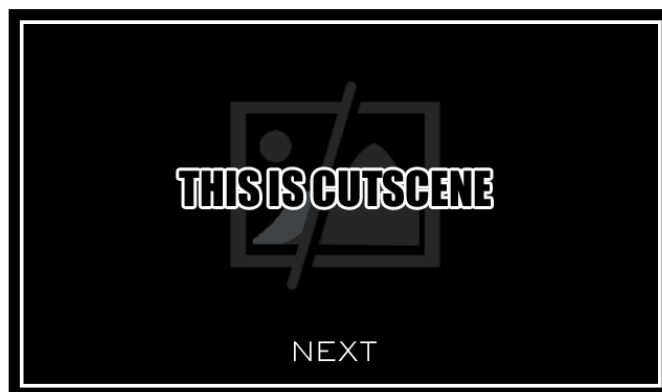
Gambar 3.11 Tampilan pada *How to Play*

Gambar 3.11 merupakan tampilan permainan ketika memilih opsi *How to Play*. Terdapat instruksi berupa cara mengendalikan pemain, menembak dan *Pause In-Game*.



Gambar 3.12 Tampilan pada *Statistics*

Pada Gambar 3.12 ini merupakan bagian terakhir dari alur *flowchart* pada Gambar 3.2. Gambar ini menunjukkan statistik berupa jumlah musuh yang berhasil dihancurkan. Terdapat opsi untuk mengatur ulang statistic tersebut.



Gambar 3.13 Tampilan pada *Cutscene* setelah memulai permainan

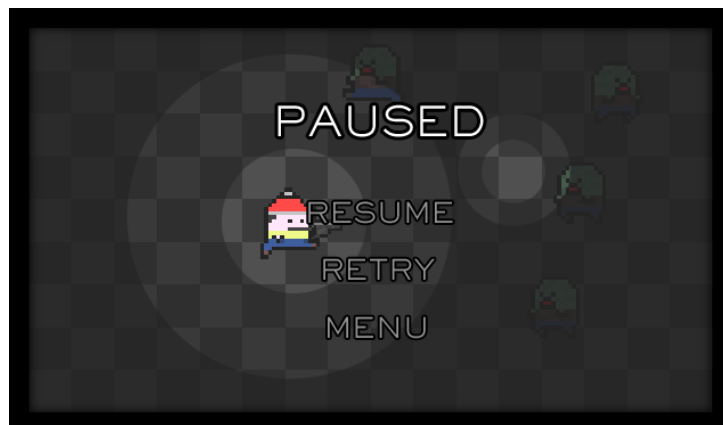
Gambar 3.13 merupakan *mockup Cutscene* setelah memilih opsi untuk memulai permainan. terdapat 7 jenis *Cutscene* satu diantaranya merupakan *Cutscene* awal cerita ketika memulai permainan dan pada kelipatan 5 tingkat

permainan akan memuat *Cutscene* awal ketika mengakses *Boss Level* dan *Cutscene* akhir ketika berhasil mengalahkan boss.



Gambar 3.14 Tampilan *Gameplay*

Gambar 3.14 merupakan tampilan dari permainan, ketika pemain berada pada sesi *gameplay*, tampilan permainan ini akan muncul. Tampilan yang muncul berupa jumlah amunisi, darah dan jumlah makanan yang dimiliki, posisi pemain, tampilan tingkat permainan yang di tutupi dengan *player vision* dan jarak senter yang dikendalikan melalui kursor mouse. Objektif pada bagian ini adalah untuk menghancurkan seluruh musuh yang ada pada tingkat permainan untuk dapat masuk tingkat selanjutnya.



Gambar 3.15 Tampilan ketika permainan dihentikan sementara

Gambar 3.15 merupakan *mockup* dari tampilan yang dihasilkan ketika pemain memilih opsi *pause* dengan menekan tombol *escape*. Terdapat opsi untuk melanjutkan permainan, memulai permainan pada tingkat awal dan kembali ke *main menu* permainan. data berupa jumlah musuh yang dihancurkan akan disimpan pada *statistic* di *main menu scene*.



Gambar 3.16 Tampilan ketika *Game Over*

Gambar 3.16 menunjukkan *mockup* berupa kondisi pemain ketika darah pemain sudah habis. Pada bagian ini, pemain akan diberikan *advice text* dari Israelmore Ayivor. Pemain dapat memulai permainan dari awal dan kembali pada *menu* awal permainan. data berupa jumlah musuh yang dihancurkan akan disimpan pada *statistic*.